POWERFUL BISR DESIGN FOR EMBEDDED SRAM WITH SELECTABLE REDUNDANCY

[1]K Naveen, [2]AMaruthi Phanindra, [3]M Bhanu Venkatesh, [4]M Anil Kumar

Dept. of Electronics and Communication Engineering,
MLR Institute of Technology,
Hyderabad, India

***Abstract*—Built in self repair (BISR) with redundancy is an powerful yield-improvement method for embedded memories. This paper proposes an powerful BISR method which consists of mainly three modules Built-in self-test (BIST) module, a Built-in Address-Analysis (BIAA) and multiplexer (MUX).The BISR is designed elastic that it can provide four operation modes to SRAM users. And here mainly each fault address is stored only once in BISR method. In BIAA module, fault addresses and redundant ones from A one-to-one mapping to achieve high yield.in fact, adding up of words, rows, columns or blocks in the SRAMs, users can select normal words as code error. The selectable redundancy brings no plenty of area and complexity and is suitable for compiler strategy. A practical SRAM with BISR circuitry is designed and introduced based on a 22nm CMOS process. Experimental results show that the BISR occupies 20% of area and can work up to 150MHZ.**

***IndexTerms*—Built-In Self-Test (BIST) Built-In Self-Repair (BISR) Multiplexer (MUX)**

I. INTRODUCTION

The area which is occupied by the embedded memories on system-on-chip is very high. It is becoming very high day –to-day. In simple words embedded memories are dominating SOC's. By this, the rows and columns are increasing. Because of this chip size is also increasing. In order to decrease this chip area we are implementing BISR method where it decreases area of the chip and gives better self-repair. And the main thing is self-repair. We do not have self-repair techniques earlier. We are implementing that know. It contains mainly three modules BIST module, BIAA module, MUX module. Altogether are used to decrease error code. Every module has different style to do their work. And to recover the faults we also implement algorithm. Where most of the faults are recovered.

II. EXISTING SYSTEM

We have SRAMs. In SRAMs mainly if we have errors, in Order to detect and correct that error external device should be connected. Here we have an external device to correct the faults also. It is a big process to detect and correct.

III. PROPOSED SYSTEM

Here we have self repair system. Where errors are repaired at self. MBIST and algorithm are used to detect error and helps in correcting code. Programmable MBIST is also used to detect.

A.BLOCK DIAGRAM:

Block diagram is as shown in figure 1

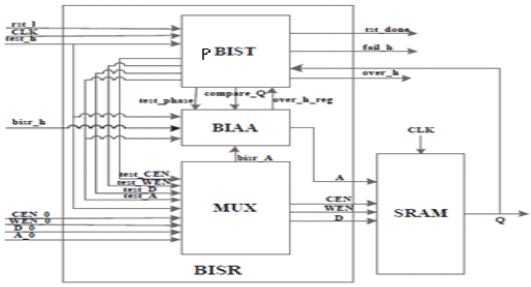


Figure.1.Block diagram BISR

B.BIST Types:

They are
1. LBIST (Logical Built in Self Test)
2. MBIST (Memory Built in Self Test)

C. Memory BIST (MBIST):

With the advent of deep-submicron VLSI technology, core-based SOC design is attracting an increasing attention. On an SOC, popular reusable cores include memories, processors, input/output circuits, etc.

Memory cores are obviously among the most universal ones - almost all system chips contain some type of embedded memory. However, to provide a low cost-cost test solution for the on-chip memory cores is not a trivial task.

D.FAULTS

Stuck – at Fault:
One or more logic values in the memory system cannot be changed i.e. the logic value of a cell or a line is always 0 or 1. For example, one or more cells are "stuck at" 1 or 0. Stuck at faults are also useful for modeling faults in other parts of the memory system.

Address decoder faults:
Any fault that affects address decoder With a certain address, no cell will be accessed.
A certain cell is never accessed.
With a certain address, multiple cells are accessed simultaneously.
A certain cell can be accessed by multiple addresses.

Coupling fault:
There exist two or more cells that are coupled. A pair of memory cells, i and j, are said to be coupled if a transition from x to y in one cell of the pair, say cell i, changes the state of the other cell, that is cell j, from 0 to 1 or from 1 to 0 i.e. a write operation to one cell changes the content of a second cell. This, of course, does not necessarily imply that a similar transition in cell j will influence cell in a similar manner.

Transition fault:
A cell or a line that fails to undergo a 0 to 1 or a 1 to 0 transition.
made whether the RAM is faulty or fault free.

E. Programmable MBIST:

Most of Memory BIST approaches concerns the programmability of the memory test algorithm. To enabling programmability of all components of memory test, test algorithm, test data, address sequence. The programmable memory

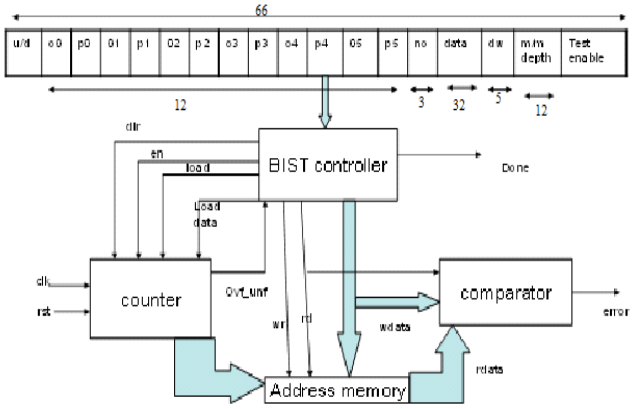
BIST proposed has several advantages:

- It enables programming both test algorithms and test data.
- It implements test algorithm programmability at low cost, by extracting the different levels of hierarchy of the test algorithm and associating a hardware block to each of them, resulting on low cost hardware.

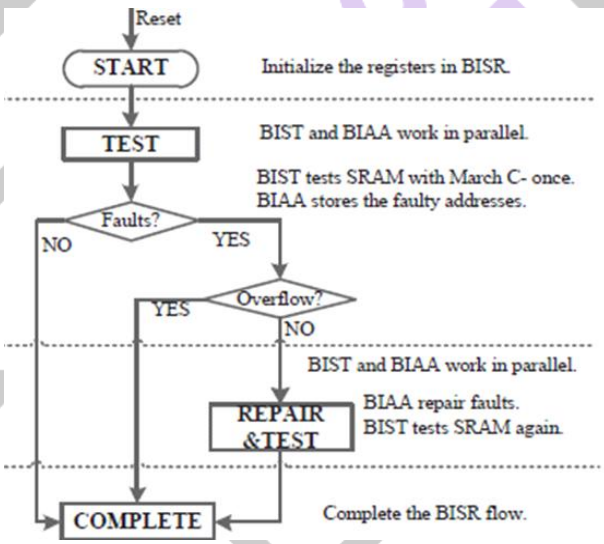
- It enables low-cost implementation of full-data programmability by adapting the transparent memory test approach in a manner that uses the memory under test for programming the test data.

The architecture for programming march test algorithms proposed in the fig. This architecture uses an instruction register specifying the current match test sequence by means of several fields indicating.

F.BIST ARCHITECTURE



G. FLOW CHART



H.IMPLEMENTATION USING MARCH ALGORITHMS:

The algorithms in most common use are the March tests. March tests have the advantage of short test time but good fault coverage. Test sequences or test algorithms for memories are known under the name of March tests. The test is "marching" through the memory. A March test consists of a sequence of March elements. A March element has a certain number of operations that must be applied to all memory cells of an array. The addressing order of a March element can be done in an up (↑), down (↓) way or (↕) if the order is not significant. A March primitive can be a write 1 (w1), write 0 (w0), read 1(r1) and read 0 (r0) that can be performed in a memory cell. The test is composed of March elements represented between (). March tests are the simplest tests (optimal) to detect most of the functional faults.
There are many March tests such as March C-, March SS, March0, March1, March LR with BDS and so on. TABLE I compares the test length, complexity and fault coverage of them. 'n' stands for the capacity of SRAM.

TABLE 1: Comparision Of March Tests

Algorithms	Test length	Complexity	Fault coverage
March C-	10n	O(n)	AF, SAF, SOF,CF
March SS	22n	O(n)	AF, SAF, SOF, CF
M0M1	4n	O(n)	SAF, SOF
March LR with BDS	23n	O(n)	AF, SAF, SOF, CF

Similarly, also observed the simulation results for,

March M 0 M 1:
{↕ (w0, r0); ↕ (w1, r1)}

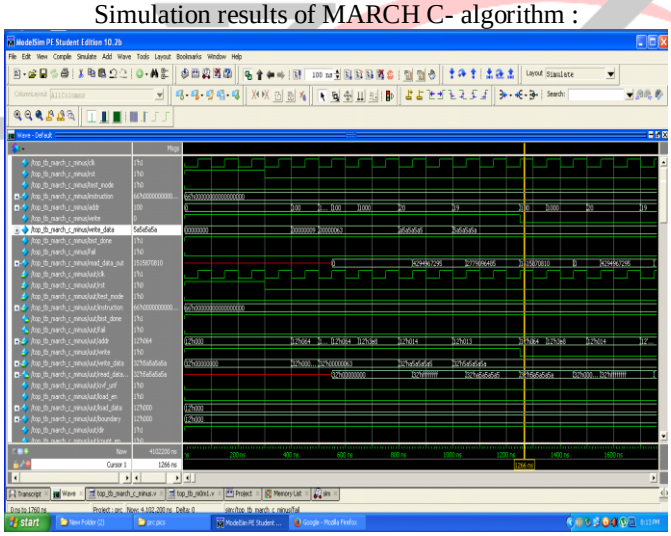
March SS(Simple Static):
{ ↕(w0); ↑(r0,r0,w0,r0,w1); ↑(r1,r1,w1,r1,w0);
↓(r0,r0,w0,r0,w1); ↓(r1,r1,w1,r1,w0); ↕(r0)}

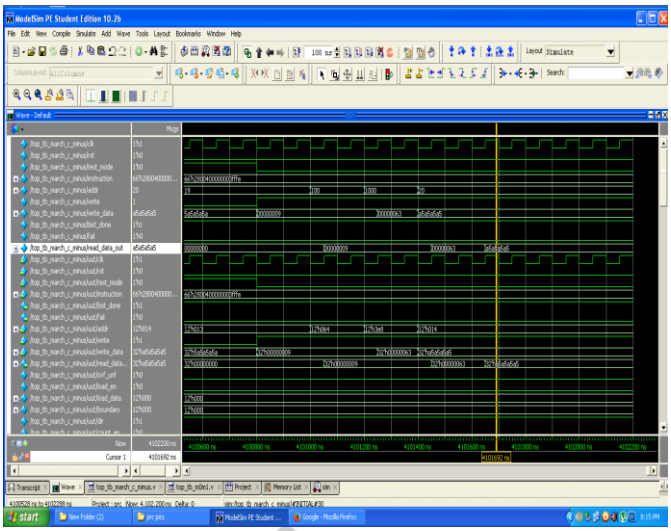
March LR(Realistic Linked) with BDS (Background Data Sequence):
{↕ (w00);DOWN (r00, w11);
UP (r11, w00, r00, r00, w11); DOWN (r11, w00);
UP(r00,w11,r11,r11,w00); DOWN(r00,w01,w10,r10);
UP(r10,w01,r01); ↑(r01)}

In above steps, "up" represents executing SRAM addresses in ascending order while "down" in descending order.

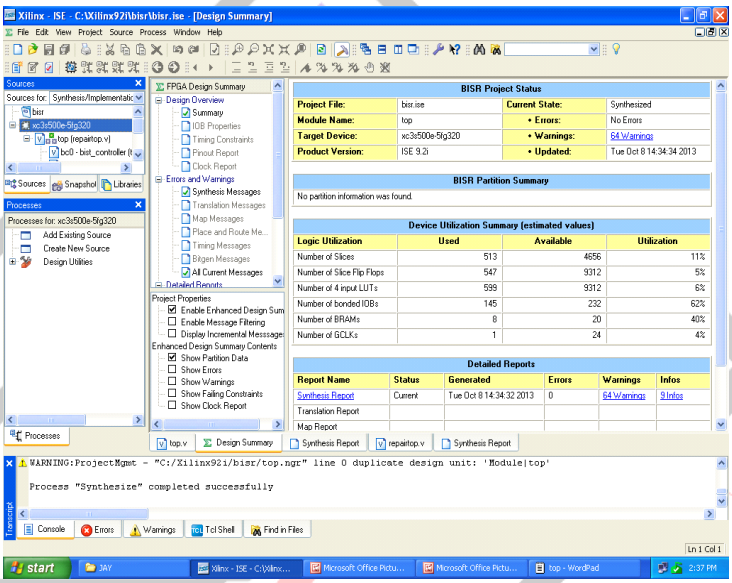
IV. SIMULATION RESULT

Simulation results of MARCH C- algorithm :





Synthesis report of Proposed BISR



V. CONCLUSION

An powerful BISR method for SRAM with programmable BIST and selectable redundancy has been presented and explained in this paper. It is designed flexible that users can choose operation modes of SRAM. The BIAA module can avoid storing fault addresses more than once and can repair fault address very fast.

As these benefits are obtained at very low area cost, the proposed memory BISR becomes highly attractive not only for test chips dedicated to new memory and/or process debug but also for integrating it into final products. The function of BISR has been verified by the post simulation. The BISR can work at up to 150MHz at the expense of 20% greater area.

VI. FUTURE SCOPE

By using BISR we can increase the performance of SRAM without any errors

ACKNOWLEDGMENT

The authors would like to express their gratitude to the management of MLR Institute of Technology for their encouragement.

REFERENCES

[1] Semiconductor Industry Association, "International technology roadmap for semiconductors (ITRS), 2003 edition," Hsinchu, Taiwan, Dec.2003.
[2] C. Stapper, A. Mclaren, and M. Dreckman, "Yield model for Productivity Optimization of VLSI Memory Chips with redundancy and Partially good Product," IBM Journal of Research and Development, Vol. 24, No. 3, pp. 398-409, May 1980.
[3] W. K. Huang, Y. H. shen, and F. lombrardi, "New approaches for repairs of memories with redundancy by row/column deletion for yield enhancement," IEEE Transactions on Computer-Aided Design, vol. 9, No. 3, pp. 323-328, Mar. 1990.
[4] P. Mazumder and Y. S. Jih, "A new built-in self-repair approach to VLSI memory yield enhancement by using neuraltype circuits," IEEE transactions on Computer Aided Design, vol. 12, No. 1, Jan, 1993.
[5] H. C. Kim, D. S. Yi, J. Y. Park, and C. H. Cho, "A BISR (built-in self- repair) circuit for embedded memory with multiple redundancies," VLSI and CAD 6th International Conference, pp. 602-605, Oct. 1999.
[6] Shyue-Kung Lu, Chun-Lin Yang, and Han-Wen Lin, "Efficient BISR Techniques for Word-Oriented Embedded Memories with Hierarchical Redundancy," IEEE ICIS-COMSAR, pp. 355-360, 2006.
[7] C. Stroud, A Designer's Guide to Built-In Self-Test,KluwerAcademicPublishers, 2002.
[8] Karunaratne. M and Oomann. B, "Yield gain with memory BISR-a case study," IEEE MWSCAS, pp. 699-702, 2009.
[9] I. Kang, W. Jeong, and S. Kang, " High-efficiency memory BISR with two serial RA stages using spare memories," IET Electron. Lett., vol. 44, no. 8, pp. 515-517, Apr. 2008.
[10] Heon-cheol Kim, Dong-soon Yi, Jin-young Park, and Chang-hyun Cho, "A BISR (Built-In Self-Repair) circuit for embedded memory with multiple redundancies," in Proc. Int. Conf. VLSI CAD, Oct. 1999.
[11] M. Sachdev, V. Zieren, and P. Janssen, " Defect detection with transient current testing and its potential for deep submicron CMOS ICs," IEEE International Test Conference, pp. 204-213, Oct. 1998.
[12] Mentor Graohics, MBISTArchitect Process Guide, Software Version8.2009_3, Aug 2009, pp. 113-116.
[13] "Efficient Built in self repair strategy for embedded sram using selectable redundancy," Huamin Cao, Ming Liu, Hong Chen, Xiang Zheng, Cong Wang and Zhihua Wang.
[14] "Memory BIST with Address Programmability," AymenFradi Michael Nicolaidis, Lorena Anghel.